

# FAULT DETECTION AND SEPARATION OF HYBRID ELECTRIC VEHICLES BASED ON KERNEL ORTHOGONAL SUBSPACE ANALYSIS

Yonghui Wang<sup>1,2</sup>, Syamsunur Deprizon<sup>1,3\*</sup>, Cong Peng<sup>1</sup>, Zhiming Zhang<sup>1</sup>

<sup>1</sup> Department of Civil Engineering, Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur, Malaysia

<sup>2</sup> College of Urban Transportation and Logistics, Shenzhen Technology University, Shenzhen, China

<sup>3</sup> Postgraduate Department, Universitas Bina Darma Palembang, Indonesia

\* deprizon@ucsiuniversity.edu.my

Driving quality and vehicles safety of hybrid electric vehicles (HEVs) are two hot-topic issues in automobile technology. Nowadays, research focuses to more intelligent and convenient HEVs fault detection methods. This paper will focus on the fault detection of HEV powertrain system with a data-driven algorithm. Orthonormal subspace analysis (OSA) is a newly proposed data-driven method which adds the ability of fault separation. Nonetheless, the linear OSA algorithm cannot effectively detect powertrain system faults, since these faults present complex nonlinear characteristics. A new kernel OSA (KOSA) method is proposed to transform the nonlinear problem into a linear problem through the mapping of kernel function and the dimensionality reduction technique of OSA. Testing results on a nonlinear model and real samples of XMQ6127AGCHEVN61 HEV show that KOSA address the nonlinear problems and it performs better than OSA and kernel principal component analysis (KPCA).

**Keywords:** fault detection, fault separation, hybrid electric vehicles, kernel function, nonlinear problem, orthonormal subspace analysis

## 1 INTRODUCTION

Hybrid electric vehicles (HEVs) have gained significant attention as a cutting-edge technology to address the pressing challenges of energy consumption, air pollution, and global warming resulting from greenhouse gas emissions [1, 2]. However, the fault detection and diagnosis (FDD) of HEV powertrain systems have not received adequate attention. Actually, a set of many powertrain devices may be extremely easy to lead to fault occurrence of HEV powertrain system [3]. The high incidence of faults in HEV powertrain systems aggravates not only energy consumption but also air pollution and global warming concerns [4].

Despite these challenges, the widespread implementation of intelligent methods in the field of HEV powertrain systems has yet to be realized. Currently, several common FDD methods are employed for HEV powertrain systems:

- The observe-based FDD method: Maintenance workers detect and locate faults based on their previous working experience. Consequently, most workers are compelled to deal with conditions that have not yet been incorporated into a mathematical model [5, 6].
- The instrument-based FDD method: Maintenance workers detect and locate faults using various instruments to measure parameters, waveforms, curves, and other relevant data. However, this approach incurs significant maintenance costs [7, 8].
- The self-diagnostic FDD method: The self-diagnostic system installed in HEVS provides cross-checks of signal levels of the Electronic Control Unit (ECU) with their reference values kept in memory. If the value of signal level exceeds that of the allowable threshold, the ECU will consider that signal as a fault one and sends a fault code to its memory. Nonetheless, the method has not been able to reach high fault detection rates (FDRs) and low false alarm rates (FARs) because of its poor anti-jamming ability [9, 10].

Compared to the FDD methods mentioned earlier, the Orthonormal Subspace Analysis (OSA) method relies solely on historical data, without the need for complex HEV models. It mitigates the drawbacks associated with traditional HEV powertrain FDD techniques. Therefore, this study adopts OSA for monitoring powertrain systems in HEVs. OSA segments the original data into two distinct categories: process variables and Key Performance Indicator (KPI) variables. Subsequently, it decomposes these categories into three subspaces, which have been validated to be orthonormal [11]. These three subspaces are identified as the quality-related subspace, the quality-unrelated process variables subspace, and the input-unrelated KPI variables subspace. Concurrently, the principal component analysis (PCA) dimensionality reduction technique [12] is employed to select the PCs of OSA, facilitating the monitoring of each subspace derived from OSA. By monitoring each subspace independently, OSA can effectively determine the presence of faults in either quality-related or quality-unrelated process variables or input-unrelated KPI variables, particularly under optimal conditions [13].

However, the traditional OSA algorithm was originally designed for monitoring linear process issues, assuming linearity in the observations. In practice, nonlinearity is prevalent in most HEV powertrain systems, affecting the

extraction of quality-related information due to nonlinear characteristics in variables. As a result, the three subspaces extracted by OSA are not truly orthonormal in the context of nonlinear processes. Artificial neural network (ANN) [14-16], which is inspired by the structure and function of biological neural networks in the brain, is a common technology for handling the nonlinearity in data. However, ANN typically requires more computational resources and time for training and execution, especially when dealing with large-scale datasets. This may result in higher computational costs. Additionally, ANN is commonly regarded as black-box models because its decision-making processes are challenging to explain. This can result in a lack of understanding of the model's internal operations. Decision trees, random forests, and gradient boosting trees, among other tree models [17-19], have abilities of capture nonlinear relationships. However, tree models are highly sensitive to minor changes in data, which can lead to unstable predictions. Small data variations may cause changes in the structure of the tree. Furthermore, in very deep trees or ensemble models, the interpretability of tree models may decrease. The kernel function, which constructs a nonlinear model mapping from the input space to the feature space, has been shown to effectively address nonlinear challenges [20, 21]. In contrast, kernel functions can effectively handle high-dimensional data without being susceptible to the curse of dimensionality because they perform inner product operations in high-dimensional space without the need to directly compute complex features of high-dimensional data [22]. Furthermore, the choice of kernel functions is relatively flexible, allowing us to not only select appropriate kernel types and parameters based on the nature of the problem, but also better adapt to various types of nonlinear relationships [23]. Most importantly, kernel function methods are typically more interpretable than ANN and tree models, as they do not produce black-box models. We can understand the mathematical properties of specific kernel functions to explain the model's behavior [24]. Furthermore, the local kernel function [25] within specific local regions makes it more suitable for addressing local nonlinearity issues. The choice between kernel function and local kernel function depends on the data characteristics and the specific problem. If the data exhibits global nonlinear structures, kernel function may be more appropriate. However, if the data contains local nonlinear structures, and we aim to capture these local structures more effectively, local kernel function may be a better choice. Therefore, we combine the kernel method with OSA in this paper, resulting in the introduction of kernel OSA (KOSA) for monitoring HEV powertrain systems.

The contributions of this work can be summarized as follows. Firstly, we propose a novel algorithm, KOSA, designed to address nonlinear fault detection challenges in HEV powertrain systems. Secondly, this work evaluates the effectiveness of the new algorithm using a nonlinear model. Additionally, KOSA has the capability to determine whether the fault is related to quality-related or quality-unrelated process variables or input-unrelated KPI variables within a nonlinear process. Finally, we conduct a comparative analysis of the fault detection rates (FDRs) and false alarm rates (FARs) for OSA, KOSA and KPCA algorithms in the context of HEV powertrain systems.

The remainder of this work is organized into five sections: Section 2 provides an overview of existing methods, including the traditional OSA algorithm and the kernel method. Section 3 introduces the KOSA algorithm for nonlinear fault detection and establishes a nonlinear model to assess the performance of the KOSA algorithm. Section 4 presents a comparative study of OSA, KOSA and KPCA algorithms based on testing with the HEV powertrain system. Finally, Section 5 offers concluding conclusions for this work.

## 2 EXISTING METHODS

### 2.1 Orthonormal Subspace Analysis

Give a set of  $n$  training samples for OSA learning and take standardized  $\mathbf{X} \in \mathbf{R}^{n \times s}$  as the  $s$  process variables. OSA retains the standard Partial Least Square identification technique [26] which introduces KPI variables, that is, one can take standardized  $\mathbf{Y} \in \mathbf{R}^{n \times r}$  as the  $r$  KPI variables. OSA can divide  $\mathbf{X}$  and  $\mathbf{Y}$  into the following three subspaces:

$$\begin{cases} \mathbf{X} = \mathbf{T}_{\text{OSA}} \mathbf{P}_{\text{OSA}} + \mathbf{E}_{\text{OSA}} \\ \mathbf{Y} = \mathbf{T}_{\text{OSA}} \mathbf{P}_{\text{OSA}} + \mathbf{F}_{\text{OSA}} \end{cases} \quad (1)$$

where  $\mathbf{T}_{\text{OSA}}$  is the score matrix extracted from the common component of  $\mathbf{X}$  and  $\mathbf{Y}$ ;  $\mathbf{P}_{\text{OSA}}$  is the transformation matrix;  $\mathbf{T}_{\text{OSA}} \mathbf{P}_{\text{OSA}}$  is the common component subspace of  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively;  $\mathbf{E}_{\text{OSA}}$  and  $\mathbf{F}_{\text{OSA}}$  are the residual subspaces of  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively.

The key principle of OSA is that the three subspaces are proved to be orthogonal, in a word, fault separation can be achieved by independently monitoring the three orthogonal subspaces. By monitoring the subspace  $\mathbf{T}_{\text{OSA}} \mathbf{P}_{\text{OSA}}$ , one can determine whether a fault locates in quality-related process variables; and by monitoring the subspace  $\mathbf{E}_{\text{OSA}}$ , one can determine whether a fault locates in quality-unrelated process variables, similarly; and one can determine whether a fault locates in the measurement of KPI variables by monitoring the subspace  $\mathbf{F}_{\text{OSA}}$ , finally. For further details, see Section 3.1.

### 2.2 The Kernel Method

The important advantage of the kernel method is mapping the original, nonlinear data into a high-dimensional linear space which is referred to as the feature space [27, 28]. That's to say, the kernel function actually builds a

nonlinear mapping model from the input space to the feature space [29, 30]. The principle of the kernel method is described in Fig. 1.

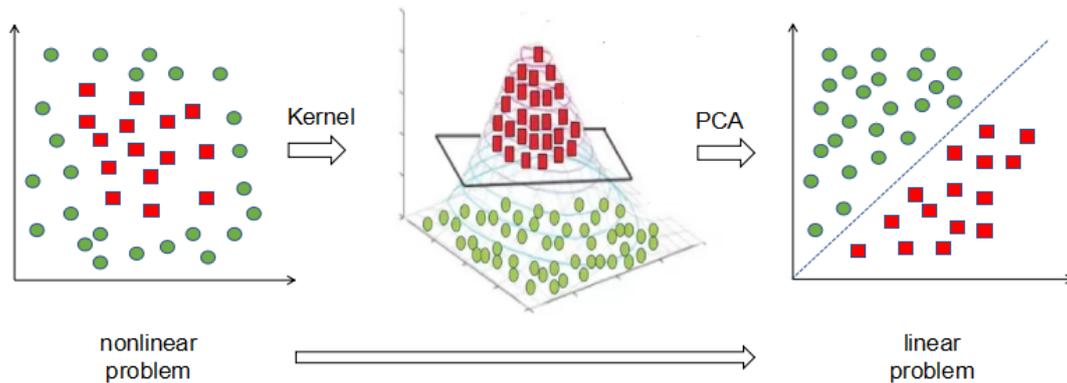


Fig. 1. Principle of the kernel method

There exists plenty of kernel functions. Several classic kernel functions are as follows:

$$\begin{cases} \tilde{K}_p(\mathbf{X}_i, \mathbf{X}_j) = \langle \mathbf{X}_i, \mathbf{X}_j \rangle^d \\ \tilde{K}_s(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\beta_0 \langle \mathbf{X}_i, \mathbf{X}_j \rangle + \beta_1) \\ \tilde{K}_r(\mathbf{X}_i, \mathbf{X}_j) = \exp(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{c}) \end{cases} \quad (2)$$

In the above equation,  $\tilde{K}_p$ ,  $\tilde{K}_s$  and  $\tilde{K}_r$  are called polynomial kernel function, sigmoid kernel function and Gaussian kernel function, respectively [31-34]. The parameters  $d > 0$ ,  $\beta_0 > 0, \beta_1 < 0$  and  $c$  can be adjusted arbitrarily.  $\mathbf{X}_i (i=1,2,\dots,n)$ ,  $\mathbf{X}_j (j=1,2,\dots,n)$  are the  $i^{th}, j^{th}$  vectors of original samples, respectively. In the given expression, " $\langle \mathbf{X}_i, \mathbf{X}_j \rangle^d$ " refers to raising the inner product of two vectors  $\mathbf{X}_i$  and  $\mathbf{X}_j$  to the power of  $d$ . The inner product is typically used for operations between matrices and vectors, representing the degree of similarity or relationship between them; " $\|\mathbf{X}_i - \mathbf{X}_j\|^2$ " represents the square of the  $L_2$ -norm of vector  $\mathbf{X}_i - \mathbf{X}_j$ . The  $L_2$ -norm calculates the square root of the sum of the squares of all elements in the vector.

Before applying the kernel method, mean-centering in the feature space should be performed as Eq. (3) shown.

$$\mathbf{K} = \tilde{\mathbf{K}} - \mathbf{1}_n \tilde{\mathbf{K}} - \tilde{\mathbf{K}} \mathbf{1}_n + \mathbf{1}_n \tilde{\mathbf{K}} \mathbf{1}_n, \quad (3)$$

where  $\mathbf{1}_n = \frac{1}{n} \begin{bmatrix} 1 & \dots & 1 \\ \dots & \dots & \dots \\ 1 & \dots & 1 \end{bmatrix} \in \mathbf{R}^{n \times n}$ ; and  $\mathbf{K}$  is the mean-centralized kernel matrix used to replace the original kernel matrix  $\tilde{\mathbf{K}}$  [35].

### 3 THE PROPOSED METHOD

#### 3.1 Kernel Orthonormal Subspace Analysis procedure

KOSA algorithm procedure is divided into two phases: offline training phase and online testing phase. Fig. 2 summarizes the procedure presented below.

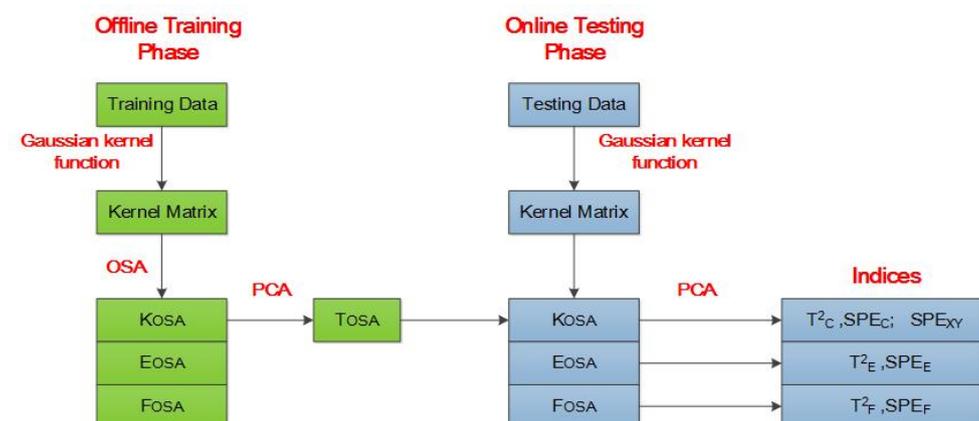


Fig. 2. Kernel orthonormal subspace analysis procedure

### A. Offline training phase:

Step 1. Suppose the offline  $n$  samples in the original input space are  $\mathbf{X}^{\text{train}} \in \mathbf{R}^{n \times s}$  (process variables) and  $\mathbf{Y}^{\text{train}} \in \mathbf{R}^{n \times r}$  (KPI variables). Take  $E(\square)$  and  $D(\square)$  as the mean and variance of data matrices. Then, data normalization technique [36] is performed as Eq. (6) shown.

$$\begin{cases} \mathbf{X} = \frac{\mathbf{X}^{\text{train}} - E(\mathbf{X}^{\text{train}})}{D(\mathbf{X}^{\text{train}})} \\ \mathbf{Y} = \frac{\mathbf{Y}^{\text{train}} - E(\mathbf{Y}^{\text{train}})}{D(\mathbf{Y}^{\text{train}})} \end{cases} \quad (4)$$

Step 2. Calculate the kernel matrices  $\tilde{\mathbf{K}}_X$  and  $\tilde{\mathbf{K}}_Y$  for  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively [22-24].

$$\begin{cases} \tilde{\mathbf{K}}_X(\mathbf{X}_i, \mathbf{X}_j) = \exp\left(-\frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{c}\right) \\ \tilde{\mathbf{K}}_Y(\mathbf{Y}_i, \mathbf{Y}_j) = \exp\left(-\frac{\|\mathbf{Y}_i - \mathbf{Y}_j\|^2}{c}\right) \end{cases} \quad (5)$$

where  $\mathbf{X}_i (i=1,2,\dots,n)$ ,  $\mathbf{X}_j (j=1,2,\dots,n)$  and  $\mathbf{Y}_i (i=1,2,\dots,n)$ ,  $\mathbf{Y}_j (j=1,2,\dots,n)$  are the  $i^{\text{th}}$ ,  $j^{\text{th}}$  vectors in the input space, respectively;

Step 3. Similar to Eq. (3), mean-centering in the feature space ought to be performed as Eq. (6) shown [22-24].

$$\begin{cases} \mathbf{K}_X = \tilde{\mathbf{K}}_X - \mathbf{1}_n \tilde{\mathbf{K}}_X - \tilde{\mathbf{K}}_X \mathbf{1}_n + \mathbf{1}_n \tilde{\mathbf{K}}_X \mathbf{1}_n \\ \mathbf{K}_Y = \tilde{\mathbf{K}}_Y - \mathbf{1}_n \tilde{\mathbf{K}}_Y - \tilde{\mathbf{K}}_Y \mathbf{1}_n + \mathbf{1}_n \tilde{\mathbf{K}}_Y \mathbf{1}_n \end{cases} \quad (6)$$

Step 4. The obtained kernel matrices  $\mathbf{K}_X$  and  $\mathbf{K}_Y$  ought to be applied to OSA.

- Calculate the common component subspace  $\mathbf{K}_{\text{OSA}}^X = \mathbf{K}_{\text{OSA}}^Y$ .

$$\begin{cases} \mathbf{K}_{\text{OSA}}^X = \mathbf{K}_Y (\mathbf{K}_Y^T \mathbf{K}_Y)^{-1} \mathbf{K}_Y^T \mathbf{K}_X \\ \mathbf{K}_{\text{OSA}}^Y = \mathbf{K}_X (\mathbf{K}_X^T \mathbf{K}_X)^{-1} \mathbf{K}_X^T \mathbf{K}_Y \end{cases} \quad (7)$$

- Calculate the residual subspaces  $\mathbf{E}_{\text{OSA}}$  and  $\mathbf{F}_{\text{OSA}}$ .

$$\begin{cases} \mathbf{E}_{\text{OSA}} = \mathbf{K}_X - \mathbf{K}_{\text{OSA}}^X \\ \mathbf{F}_{\text{OSA}} = \mathbf{K}_Y - \mathbf{K}_{\text{OSA}}^Y \end{cases} \quad (8)$$

- KPCA decomposition is used to extract the PCs. Then, one can obtain the score matrix extracted from the common component, as Eq. (9) shown.

$$\begin{cases} \mathbf{K}_{\text{OSA}}^X = \mathbf{T}_{\text{OSA}}^X \mathbf{P}_{\text{OSA}}^X + \mathbf{E}_1 \\ \mathbf{K}_{\text{OSA}}^Y = \mathbf{T}_{\text{OSA}}^Y \mathbf{P}_{\text{OSA}}^Y + \mathbf{E}_2 \end{cases} \quad (9)$$

where  $\mathbf{T}_{\text{OSA}}^X = \mathbf{T}_{\text{OSA}}^Y$  are the score matrices extracted from the common component;  $\mathbf{E}_1$  and  $\mathbf{E}_2$  are residual matrices. It is worth mentioning that the PCs are chosen by the PCA method, and the value of cumulative contribution rate ought to be calculated by the PCA method [12].

- Offline thresholds calculation:

$\mathbf{K}_{\text{OSA}}^X$  or  $\mathbf{K}_{\text{OSA}}^Y$ ,  $\mathbf{E}_{\text{OSA}}$  and  $\mathbf{F}_{\text{OSA}}$  ought to be monitored by the PCA method and then generate  $T^2$  and SPE monitoring indices, that is,  $T_C^2$ ,  $T_E^2$ ,  $T_F^2$ ,  $\text{SPE}_C$ ,  $\text{SPE}_E$ ,  $\text{SPE}_F$  and  $\text{SPE}_{XY}$ . The thresholds ought to refer to the PCA method [12].

What's more, if the relationship between  $\mathbf{X}$  and  $\mathbf{Y}$  fails,  $\mathbf{K}_{\text{OSA}}^X$  will not amount to  $\mathbf{K}_{\text{OSA}}^Y$ . Therefore, a following index is constructed to monitor the changing relationship between  $\mathbf{X}$  and  $\mathbf{Y}$ :

$$\text{SPE}_{XY} = \|\mathbf{T}_{\text{OSA}}^X - \mathbf{T}_{\text{OSA}}^Y\|^2 \quad (10)$$

whose threshold ought to refer to the PCA method [12], as well.

### B. Online testing phase:

Step 1. Suppose the online  $m$  samples in the original input space are  $\mathbf{X}^{\text{test}} \in \mathbf{R}^{m \times s}$  and  $\mathbf{Y}^{\text{test}} \in \mathbf{R}^{m \times r}$ . Similarly, data normalization technique ought to be performed as Eq. (11) shown.

$$\begin{cases} \mathbf{X}^{\text{new}} = \frac{\mathbf{X}^{\text{test}} - E(\mathbf{X}^{\text{train}})}{D(\mathbf{X}^{\text{train}})} \\ \mathbf{Y}^{\text{new}} = \frac{\mathbf{Y}^{\text{test}} - E(\mathbf{Y}^{\text{train}})}{D(\mathbf{Y}^{\text{train}})} \end{cases} \quad (11)$$

Note that the mean and variance of the offline samples are still used for data normalization technique in online testing phase.

Step 2. Calculate the kernel matrices for  $\mathbf{X}^{\text{new}}$  and  $\mathbf{Y}^{\text{new}}$ , respectively.

Step 3. Mean-centering in the feature space ought to be performed similar to Eq. (6).

Step 4. Calculate the common component subspace and residual subspaces similar to Eq. (9).

Step 5. Online monitor indices calculation: calculate the monitor indices  $T_C^2$ ,  $T_E^2$ ,  $T_F^2$ ,  $SPE_C$ ,  $SPE_E$ ,  $SPE_F$  and  $SPE_{XY}$ . If the value of any monitor index exceeds that of its threshold, there will exist a fault [12].

### 3.2 Nonlinear model application

In this Section, a nonlinear model was applied to test the nonlinear ability of KOSA. The simulated process was given by Eq. (12):

$$\begin{cases} \mathbf{x}_1 = e^{C_1} + C_1 \sin(C_2) + 0.01s_1 \\ \mathbf{x}_2 = C_1^3 + e^{C_1} + 0.01s_2 \\ \mathbf{x}_3 = C_1 C_2 + 0.01s_3 \\ \mathbf{x}_4 = (C_1 + C_2)^2 + 0.01s_4 \\ \mathbf{x}_5 = C_1 \sin(C_1) + C_1^2 + 0.01s_5 \\ \mathbf{x}_6 = C_1^3 + C_2^2 + 0.01s_6 \\ \mathbf{x}_7 = C_1 + 0.01s_7 \\ \mathbf{x}_8 = C_1^2 + C_2^2 + 0.01s_8 \\ \mathbf{x}_9 = C_2 + 0.01s_9 \\ \mathbf{x}_{10} = C_1 + C_2 + 0.01s_{10} \\ \mathbf{y}_1 = C_2^2 + C_3^3 + 0.01s_{11} \\ \mathbf{y}_2 = C_3 + 0.01s_{12} \end{cases} \quad (12)$$

where variables  $C_k$  ( $k=1,2,3$ ) followed a uniform distribution in interval  $-1,1$ ; variables  $s_k$  ( $k=1,2,\dots,12$ ) represented the Gaussian process noise; Certainly,  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{10}]$  acted as process variables while  $[\mathbf{y}_1, \mathbf{y}_2]$  acted as KPI variables. About 1,000 normal samples were obtained. The code was implemented using the MATLAB platform, and it can be reproduced based on the algorithm steps outlined in subsection 3.1. In this paper, Gaussian kernel function with kernel width  $c=3000$  is selected because it was found that  $c=3000$  can lead to the optimal performance for KOSA.

- Fault 1: A step fault with an amplitude of 3 in  $C_1$ , for variables  $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{10}]$ . Obviously, Fault 1 located in quality-unrelated process variables. The fault detection rates (FDRs) and false alarm rates (FARs) of Fault 1 are shown in Table 1.

Table 1. FDRs and FARs of Fault 1

	$T_C^2$	$SPE_C$	$T_E^2$	$SPE_E$	$T_F^2$	$SPE_F$	$SPE_{XY}$
FARs	0%	0%	0%	0%	0%	0%	0%
FDRs	6%	5.5%	62%	98%	4.5%	0%	91.25%

From the results shown in Table 1, the FDRs of  $T_E^2$ ,  $SPE_E$  and  $SPE_{XY}$  were extremely higher than the FDRs of others so it proved that the fault occurred in quality-unrelated process variables. Fig.3 shows the Fault 1 detection diagram of  $T_E^2$ ,  $SPE_E$  and  $SPE_{XY}$ , where the blue line represented values of the indices and the red one represented values of the thresholds.

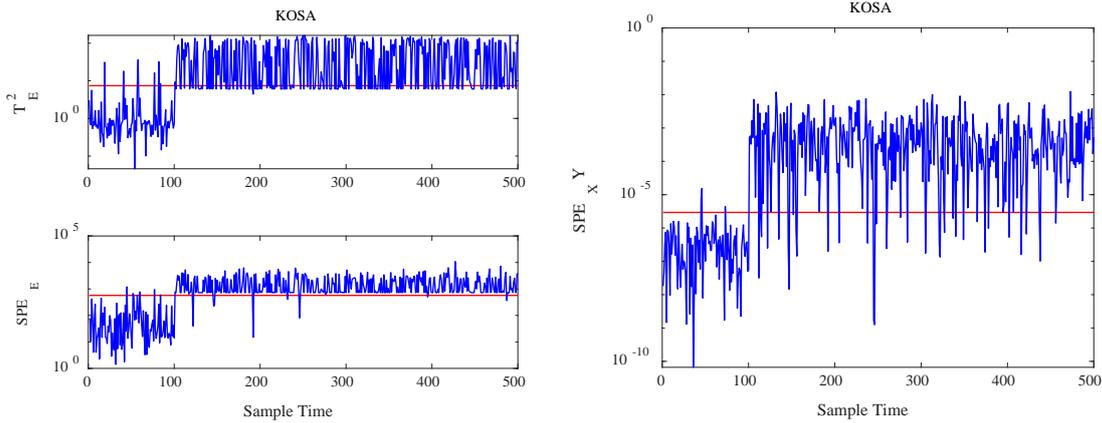


Fig. 3. Fault 1 detection diagram of  $T_E^2$ ,  $SPE_E$  and  $SPE_{XY}$

- Fault 2: A step fault with an amplitude of 3 in  $C_2$ , for all the variables. Similarly, Fault 2 located in quality-related process variables. The FDRs and FARs of Fault 2 are shown in Table 2.

Table 2. FDRs and FARs of Fault 2

	$T_C^2$	$SPE_C$	$T_E^2$	$SPE_E$	$T_F^2$	$SPE_F$	$SPE_{XY}$
FARs	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>
FDRs	<b>94.5%</b>	<b>83.5%</b>	0%	13%	1.5%	57.25%	<b>97.5</b>

From the results shown in Table 2, the FDRs of  $T_C^2$ ,  $SPE_C$  and  $SPE_{XY}$  were extremely high so it proved that the fault occurred in quality-related process variables and the relationship between  $X$  and  $Y$  failed. Fig.4 shows the FDRs of  $T_C^2$ ,  $SPE_C$  and  $SPE_{XY}$ , where the blue line represented values of the indices and the red one represented values of the thresholds.

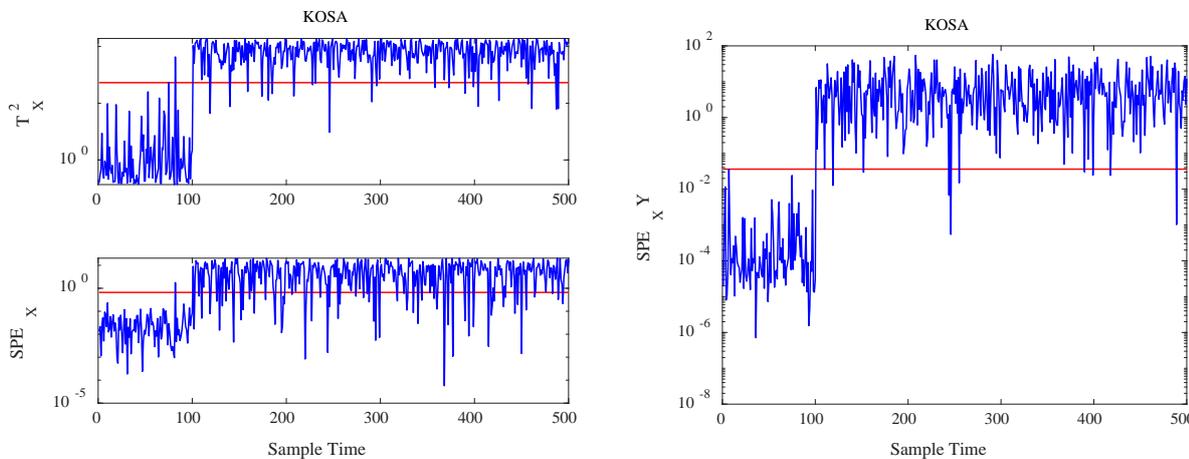


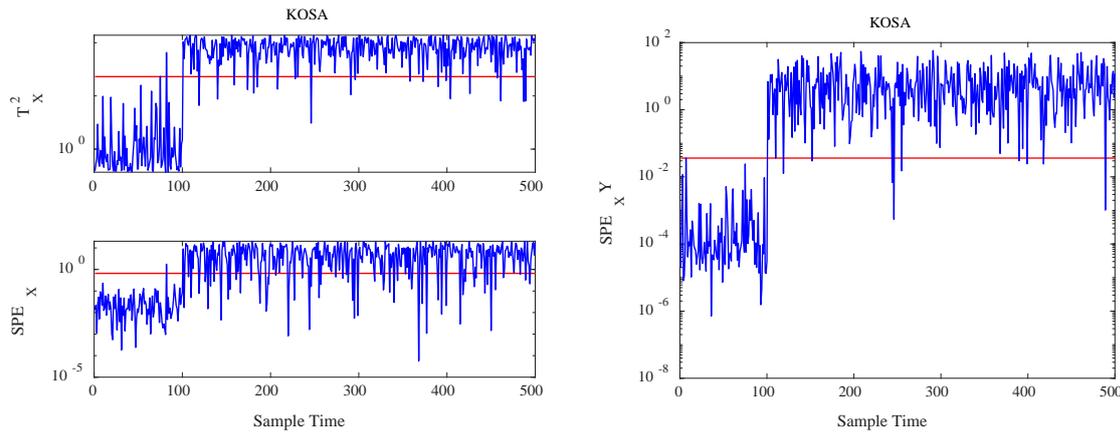
Fig. 4. Fault 2 detection diagram of  $T_C^2$ ,  $SPE_C$  and  $SPE_{XY}$

Fault 3: A step fault with an amplitude of 3 in  $C_3$ , for variables  $[y_1, y_2]$ . Finally, Fault 3 located in the measurement of KPI variables. The FDRs and FARs of Fault 3 are shown in Table 3.

Table 3. FDRs and FARs of Fault 3

	$T_C^2$	$SPE_C$	$T_E^2$	$SPE_E$	$T_F^2$	$SPE_F$	$SPE_{XY}$
FARs	0%	0%	0%	0%	0%	0%	0%
FDRs	0%	2.75%	1.5%	0.5%	<b>83.5%</b>	<b>99%</b>	<b>95%</b>

From the results shown in Table 3, the FDRs of  $T_F^2$ ,  $SPE_F$  and  $SPE_{XY}$  were especially high so it proved that the fault occurred in the measurement of KPI variables and the relationship between  $X$  and  $Y$  failed. Fig.5 shows the FDRs of  $T_F^2$ ,  $SPE_F$  and  $SPE_{XY}$ , where the blue line represented values of the indices and the red one represented values of the thresholds.

Fig. 5. The Fault 3 detection diagram of  $T_F^2$ ,  $SPE_F$  and  $SPE_{XY}$ 

## 4 TESTING WITH HEVS POWERTRAIN SYSTEM

### 4.1 Data selection

The approach presented in this paper had been tested in XMQ6127AGCHEVN61 HEV. Some variables of the powertrain system were selected and shown as follows:

Table 4. Definition of Variables

Variable number	Variable source	Variable name	Process variable/ KPI
1	Engine Control Module	The actual engine torque (Nm)	Process variable
2		The desired engine torque (Nm)	Process variable
3		The actual engine speed (r/s)	Process variable
4		The desired engine speed (r/s)	Process variable
5		Water temperature (°C)	Process variable
6		Engine oil temperature (°C)	Process variable
7	Motor control unit	The actual motor torque (Nm)	Process variable
8		The desired motor torque (Nm)	Process variable
9		The actual motor speed (r/s)	Process variable
10		The desired motor speed (r/s)	Process variable
11		Motor temperature (°C)	Process variable
12		Motor current (A)	Process variable
13	Electronic Stability Program	Accelerator pedal opening (%)	Process variable
14		<b>The actual vehicle speed (km/h)</b>	<b>KPI</b>
15		The desired vehicle speed (km/h)	Process variable
16		The actual vehicle torque (Nm)	Process variable
17		The desired vehicle torque (Nm)	Process variable

This paper mainly studied the powertrain performance of HEVs, so the actual vehicle speed (the 14<sup>th</sup> variable) was selected as the KPI variable. The code was implemented using the MATLAB platform, and it can be reproduced based on the algorithm steps outlined in subsection 3.1. In this paper, Gaussian kernel function with kernel width  $c = 3000$  is selected because it was found that  $c=3000$  can lead to the optimal performance for KOSA.

### 4.2 Testing

The real samples of XMQ6127AGCHEVN61 HEV were collected when the accelerator was not functioning. A maximum of 1,500 normal road test samples were used in the offline training phase; a maximum of  $m = 1100$  non-operational accelerator samples were selected in the online testing phase, and all faults were introduced from the  $m_0^{\text{th}} = 100^{\text{th}}$  sample up to the last on. The FAR and the FDR were calculated as Eq. (13) shown [11].

$$\begin{cases} FAR = \frac{\text{num\_F}_0}{m_0} \times 100\% \\ FDR = \frac{\text{num\_F}}{m - m_0} \times 100\% \end{cases} \quad (13)$$

where num\_F<sub>0</sub> is the number of fault samples in the first m<sub>0</sub> samples; and num\_F is the number of fault samples in the last m - m<sub>0</sub> samples. KOSA, OSA algorithms were used for this comparison study, and the results of FARs and FDRs are shown in Table 5.

Table 5. FARs and FDRs of OSA and KOSA

		T <sub>C</sub> <sup>2</sup>	SPE <sub>C</sub>	T <sub>E</sub> <sup>2</sup>	SPE <sub>E</sub>	T <sub>F</sub> <sup>2</sup>	SPE <sub>F</sub>	SPE <sub>XY</sub>
FARs	OSA	0%	0%	0%	0%	0%	0%	0%
	KOSA	0%	0%	0%	0%	0%	0%	0%
FDRs	OSA	54.5%	0%	0%	35%	0%	9.4%	37.9%
	KOSA	100%	41%	0%	100%	0%	56.8%	99.9%

From the analysis of Table 5, faults mostly occurred in quality-related and quality-unrelated process variables, and the relationship between X and Y failed. Moreover, almost 50% samples were faults which occurred in the measurement of KPI variables. It's obvious that the FDRs of the OSA algorithm were lower than that of the KOSA algorithm. Fig.6 shows the detection chart of the KOSA algorithm, where the blue line showed values of indices and the red one showed values of thresholds.

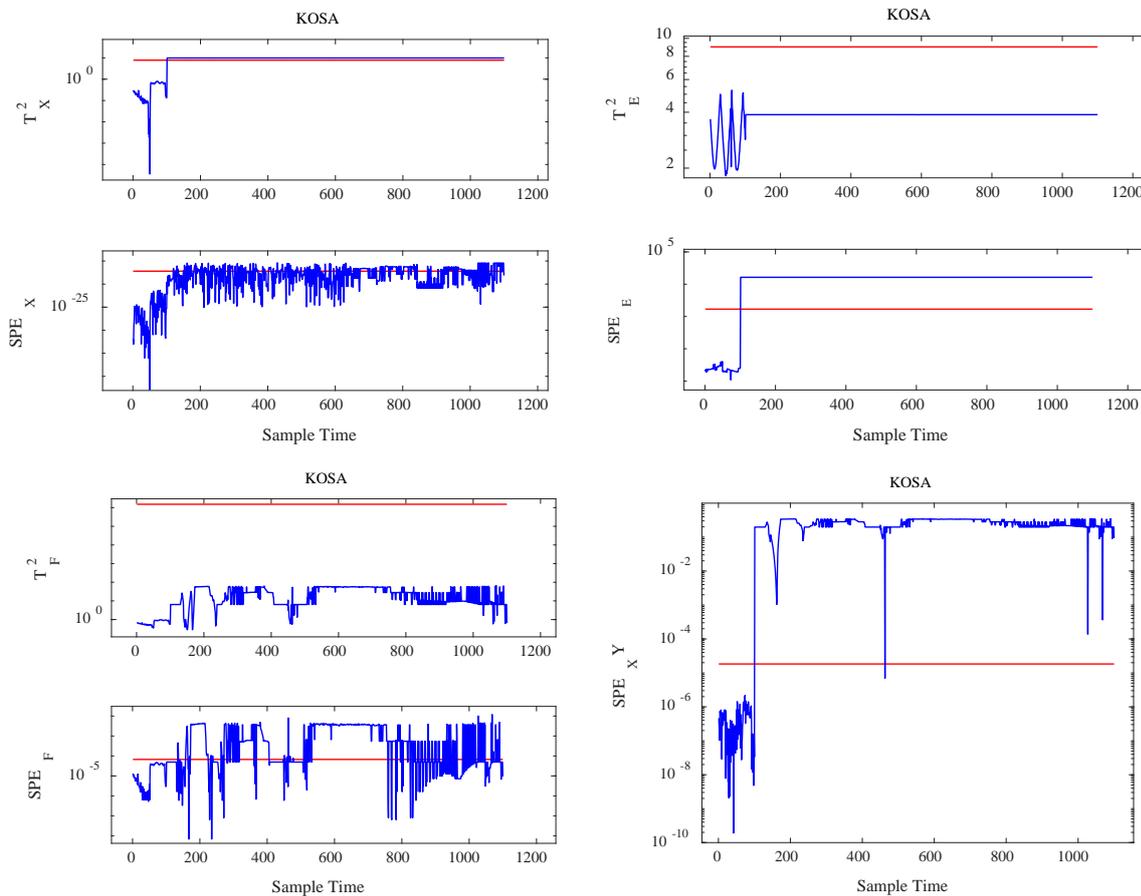


Fig. 6. HEV powertrain fault detection chart of KOSA

As a KPI-related algorithm, KOSA, when combined with a kernel function, can achieve better monitoring of nonlinear data. To substantiate this, we conducted comparative simulations with KPCA [20] before studying KOSA. A maximum of m' = 1900 non-operational accelerator samples were selected in the online testing phase, and all faults were introduced from the m<sub>0</sub><sup>th</sup> = 501<sup>th</sup> sample up to the last one. Table 6 and Fig.7 show the performance of the KPCA algorithm.

Table 6. FARs and FDRs of KPCA

	$T^2$	SPE
FARs	0%	0%
FDRs	38.5%	55.79%

From the analysis of Table 6, while KPCA demonstrates commendable FARs, its FDR remains below 60%. The reason behind this is that KPCA, as a KPI-unrelated algorithm, treats all variables uniformly and does not take into account the impact of KPI variables. This lack of specificity hinders its effectiveness in monitoring the powertrain system.

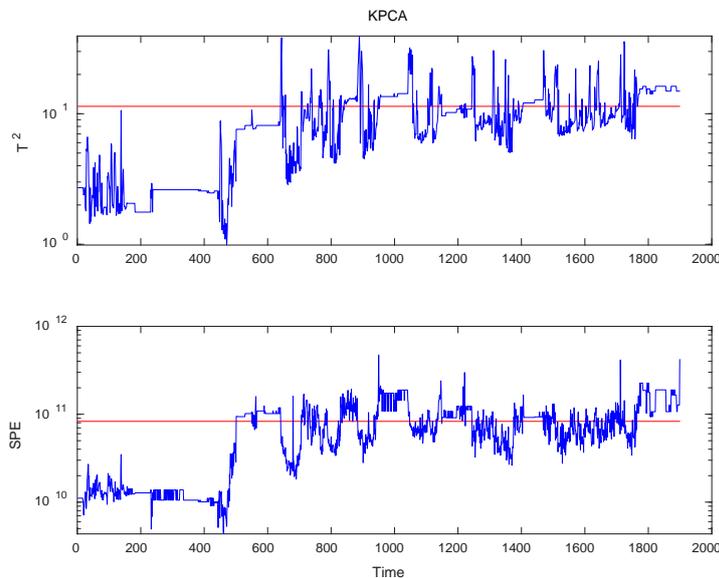


Fig. 7. HEV powertrain fault detection chart of KPCA

From the analysis of Figure 7, starting from the 650<sup>th</sup> sample, the blue line which represents the values of the fault statistic in the  $T^2$  statistical chart occasionally surpassed the red threshold line. However, it is evident that the blue line frequently dropped below the red line, indicating that KPCA lost its ability to distinguish between normal and faults. Similarly, in the SPE statistical chart, from the 500<sup>th</sup> sample onward, the blue line began to trend above the red line, but it still failed to effectively distinguish between normal and faults. As a result, KOSA is better suited for monitoring HEV powertrain systems compared to KPCA.

## 5 CONCLUSION

This paper focuses on fault detection and separation in nonlinear HEV powertrain systems using a data-driven algorithm called KOSA, an enhanced version of the OSA algorithm. Compared to other nonlinear methods, KOSA offers enhanced interpretability: KOSA has the capability to transform the nonlinear high-dimensional variables into a linear framework, allowing for their monitoring using linear OSA method. Testing results, obtained from a nonlinear model and a real HEV powertrain system, demonstrate that the KOSA algorithm excels at extracting features from nonlinear data and accurately determining whether the fault resides in quality-related process variables, quality-unrelated process variables, or KPI variables. However, KOSA is a promising algorithm with the potential for further development. However, there are certain limitations that warrant future work:

- KOSA experiences a noticeable decrease in processing speed when dealing with data samples exceeding 1,000.
- While KOSA is well-suited for handling strongly nonlinear data, it is less effective in addressing dynamic issues. Readers are encouraged to explore improvements for dynamic problems in future research, such as combining OSA with dynamic methods [13].
- Due to the real-time processing of data in the online phase, KOSA utilizes mean and variance from the offline phase for data standardization. Consequently, when establishing an online KOSA model, there may be model errors associated with this approach.

## 6 REFERENCES

- [1] Liu, F., Liu, B. (2023). Fault mode detection of a hybrid electric vehicle by using support vector machine. *Energy Reports*, vol. 9, pp. 137-148, DOI: <https://doi.org/10.1016/j.egy.2023.04.328>.

- [2] Elsayed, M.E., Hamad, M.S, (2021). Modeling and Analysis of PMSM under Regenerative Braking Operations with Fault-Tolerant for EV/HEV Applications. *2021 31st International Conference on Computer Theory and Applications (ICCTA)*, pp. 112-119.
- [3] Cammalleri, M., Di Dio, V. (2023). Special Issue on Frontiers in Hybrid Vehicles Powertrain. *Applied Sciences*, vol. 13, no. 11, p. 6367, DOI: <https://doi.org/10.3390/app13116367>.
- [4] Anselma, P.G.J.E. (2021). Optimization-driven powertrain-oriented adaptive cruise control to improve energy saving and passenger comfort. *Energies*, vol. 14, no. 10, p. 2897, DOI: <https://doi.org/10.3390/en14102897>.
- [5] Jiang, W., Abolfazli, E., (2021). Observer-based control for vehicle platooning with a leader of varying speed. *2021 European Control Conference (ECC)*, pp. 2-9.
- [6] Li, M., Bai, C. (2021). Application of Electronic Diagnosis Technology in New Energy Vehicle Maintenance. *Earth and Environmental Science*, vol. 680, no. 1, p. 012043, DOI: 10.1088/1755-1315/680/1/012043.
- [7] Chen, J. (2021). Model-based validation of diagnostic software with application in automotive systems. *IET Cyber-Systems and Robotics*, vol. 3, no. 2, pp. 140-149, DOI: <https://doi.org/10.1049/csy2.12016>.
- [8] Kobti, Z. and El-Ghaname, J. (2023). Using Knowledge Graph Embedding for Fault Detection: A Case Study in Electric Vehicle Parts Assembly. *The International FLAIRS Conference Proceedings*, vol. 36.
- [9] Fu, Q. (2022). A DSP-Controlled Permanent Magnet Synchronous Motor Control System for Hybrid Vehicles. *International Journal of Antennas and Propagation*, vol. 2022, p. 1996502, DOI: <https://doi.org/10.1155/2022/1996502>.
- [10] Wan, P., Liu, B. (2023). Engine modelling architecture study for hybrid electric vehicle diagnosis application. *Energy*, vol. 282, p. 128408, DOI: <https://doi.org/10.1016/j.energy.2023.128408>.
- [11] Lou, Z., Wang, Y. (2022). A novel multivariate statistical process monitoring algorithm: Orthonormal subspace analysis. *Automatica*, vol. 138, p. 110148, DOI: <https://doi.org/10.1016/j.automatica.2021.110148>.
- [12] Lou, Z., Hao, W. (2023). Thermal-Imaging-Based PCA Method for Monitoring Process Temperature. *Processes*, vol. 11, no. 2, p. 589, DOI: <https://doi.org/10.3390/pr11020589>.
- [13] Hao, W., Lu, S. (2023). A Novel Dynamic Process Monitoring Algorithm: Dynamic Orthonormal Subspace Analysis. *Processes*, vol. 11, no. 7, p. 1935, DOI: <https://doi.org/10.3390/pr11071935>.
- [14] Kanwisher, N., Khosla, M. (2023). Using artificial neural networks to ask 'why' questions of minds and brains. *Trends in Neurosciences*, vol. 46, no. 3, pp. 240-254, DOI: <https://doi.org/10.1016/j.tins.2022.12.008>.
- [15] Li, Z., Wang, Y. (2022). Neural Component Analysis for Key Performance Indicator Monitoring. *ACS omega*, vol. 7, no. 42, pp. 37248-37255, DOI: <https://doi.org/10.1021/acsomega.2c03515>.
- [16] Wang, Y., Lou, Z., (2021). Nonlinear PLS with Neural Component Analysis Structure. *2021 CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS)*, pp. 1-4, DOI: 10.1109/SAFEPROCESS52771.2021.9693603.
- [17] Costa, V.G., Pedreira, C.E. (2022). Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765-4800, DOI: <https://doi.org/10.1007/s10462-022-10275-5>.
- [18] Hu, J., Szymczak, S. (2023). A review on longitudinal data analysis with random forest. *Briefings in Bioinformatics*, vol. 24, no. 2, p. bbad002, DOI: <https://doi.org/10.1093/bib/bbad002>.
- [19] Brophy, J., Hammoudeh, Z. (2022). Adapting and Evaluating Influence-Estimation Methods for Gradient-Boosted Decision Trees. *Journal of Machine Learning Research*, vol. 24, pp. 1-48, DOI: 10.48550/arXiv.2205.00359.
- [20] Li, D., Wong, W., (2021). Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and K-means. *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 551-559, DOI: 10.1109/DSA52907.2021.00081.
- [21] Yan, X., Guan, T. (2021). Novel double layer BiLSTM minor soft fault detection for sensors in air-conditioning system with KPCA reducing dimensions. *Journal of Building Engineering*, vol. 44, p. 102950, DOI: <https://doi.org/10.1016/j.jobe.2021.102950>.
- [22] Pande, C.B., Kushwaha, N.L. (2023). Comparative assessment of improved SVM method under different kernel functions for predicting multi-scale drought index. *Water Resources Management*, vol. 37, no. 3, pp. 1367-1399, DOI: <https://doi.org/10.1007/s11269-023-03440-0>.
- [23] Cao, K., Dong, F. (2023). Infrared radiation denoising model of "sub-region-Gaussian kernel function" in the process of sandstone loading and fracture. *Infrared Physics & Technology*, vol. 129, p. 104583, DOI: <https://doi.org/10.1016/j.infrared.2023.104583>.
- [24] Liu, A., Zhao, D. (2023). A data classification method based on particle swarm optimisation and kernel function extreme learning machine. *Enterprise Information Systems*, vol. 17, no. 3, p. 1913764, DOI: <https://doi.org/10.1080/17517575.2021.1913764>.
- [25] Fang, H., Tao, W. (2022). Nonlinear dynamic process monitoring based on two-step dynamic local kernel principal component analysis. *Processes*, vol. 10, no. 5, p. 925, DOI: <https://doi.org/10.3390/pr10050925>.

- [26] Kono, S., Sato, M. (2021). Partial least squares structural equation modeling (PLS-SEM) analysis for social and management research: a literature review. *Journal of Leisure Research*, vol. 54, no. 3, pp. 309-329, DOI: <https://doi.org/10.1080/00222216.2022.2066492>.
- [27] Allegrini, F., Olivieri, A. C. (2023). Two sides of the same coin: kernel partial least-squares (KPLS) for linear and non-linear multivariate calibration. *Talanta Open*, vol. 7, p. 100235, DOI: <https://doi.org/10.1016/j.talo.2023.100235>.
- [28] Guo, H., Sun, J. (2023). Quality-Related Process Monitoring and Diagnosis of Hot-Rolled Strip Based on Weighted Statistical Feature KPLS. *Sensors*, vol. 23, no. 13, p. 6038, DOI: <https://doi.org/10.3390/s23136038>.
- [29] Guo, H., Sun, J. (2023). Fault Diagnosis of Check Valve Based on KPLS Optimal Feature Selection and Kernel Extreme Learning Machine. *Coatings*, vol. 12, no. 9, p. 1320, DOI: <https://doi.org/10.3390/coatings12091320>.
- [30] Yuan, P., Zhao, L. (2022). KPLS-based Mach number prediction for multi-mode wind tunnel flow system. *Processes*, vol. 10, no. 9, p. 1718, DOI: <https://doi.org/10.3390/pr10091718>.
- [31] Zhang, K., Zhang, K. (2023). Prediction of gas explosion pressures: A machine learning algorithm based on KPCA and an optimized LSSVM. *Journal of Loss Prevention in the Process Industries*, vol. 83, p. 105082, DOI: <https://doi.org/10.1016/j.jlp.2023.105082>.
- [32] Wang, C., Zheng, J. (2022). Deeppipe: Operating condition recognition of multiproduct pipeline based on KPCA-CNN. *Journal of Pipeline Systems Engineering and Practice*, vol. 13, no. 2, p. 04022006, DOI: [https://doi.org/10.1061/\(ASCE\)PS.1949-1204.0000641](https://doi.org/10.1061/(ASCE)PS.1949-1204.0000641).
- [33] Arowolo, M. O., Ogundokun, R. O. (2022). Machine learning approach using KPCA-SVMs for predicting COVID-19. *Healthcare informatics for fighting COVID-19 and future epidemics*, pp. 193-209, DOI: [https://doi.org/10.1007/978-3-030-72752-9\\_10](https://doi.org/10.1007/978-3-030-72752-9_10).
- [34] Duan, Y., Huang, F. (2023). Intelligent spectrum sensing algorithm for cognitive internet of vehicles based on KPCA and improved CNN. *Peer-to-Peer Networking and Applications*, vol. 16, no. 5, pp. 2202-2217, DOI: <https://doi.org/10.1007/s12083-023-01501-0>.
- [35] Yang, F., Ma, Z. (2022). Image classification with parallel KPCA - PCA network. *Computational Intelligence*, vol. 38, no. 2, pp. 397-415, DOI: <https://doi.org/10.1111/coin.12503>.
- [36] Huang, L., Qin, J. (2023). Normalization techniques in training DNNs: Methodology, analysis and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10173-10196, DOI: 10.1109/TPAMI.2023.3250241.
- [37] Pan, C., Tan, J. (2021). Prediction intervals estimation of solar generation based on gated recurrent unit and kernel density estimation. *Neurocomputing*, vol. 453, pp. 552-562, DOI: <https://doi.org/10.1016/j.neucom.2020.10.027>.

Paper submitted: 04.08.2023.

Paper accepted: 07.11.2023.

This is an open access article distributed under the CC BY 4.0 terms and conditions